CLAIMS

What is claimed is:

1. A system that maps a first construct to a second construct, comprising:

a bank that stores at least one of a set of suppress field labels and a set of introduce field labels; and

a mapping component that utilizes at least one of a suppress field label and an introduce field label to facilitate mapping the first construct to the second construct.

2. The system of claim 1, the first construct is a named or an anonymous construct and the second construct is a named or an anonymous construct, and the mapping comprises one of transforming a first named construct to a second named construct; the first named construct to a second anonymous construct; a first anonymous construct to a second named construct; and the first anonymous construct to a second anonymous construct.

3. The system of claim 1, the first construct is one of a markup language construct, an object orientated language construct, a relational construct and a user interface construct, and the second construct is one of a markup language construct, an object orientated language construct, a relational construct and a use interface construct.

4. The system of claim 3, the markup language construct is one of an XML and a CLR construct, the object oriented language construct is one of a C++, C#, Java and Visual Basic construct, and the relational construct is a SQL construct.

5. The system of claim 1, the mapping is isomorphic.

6. The system of claim 1, further comprising a mapping file that provides one or more of a default mapping, user customized mapping, and a mediating schema that facilitates mapping the first construct to the second construct.

7. The system of claim 6, the user customized mapping defines the construct structure to suppress and introduce.

8. The system of claim 6, the user customized mapping comprises at least one of annotating types and annotating schema.

9. The system of claim 6, the default mapping is based on one or more of a heuristic, an inference, a probability and machine learning.

10. The system of claim 6, the meditating schema transforms constructs to an intermediate representation at least one of before, during and after transforming the first construct.

11. The system of claim 1, the first construct is a complex or simple construct and the second construct is a complex or simple construct.

12. The system of claim 1, the mapping comprises one or more of serializing an instance of the first construct to the second construct; deserializing an instance of the first construct to the second construct; persisting the first construct to the second construct; restoring the first construct from the second construct; publishing the first construct in the second construct; shredding the first construct from the second construct; and binding the first construct to the second construct.

13. A method that transforms constructs between domains, comprising:

receiving a construct;

obtaining a mapping associated with the construct; and

employing the mapping to transform the construct from a first domain to a second domain.

14. The method of claim 13, further comprising transforming one of a named construct to a different named construct; the named construct to an anonymous construct; an anonymous construct to a different anonymous construct; and an anonymous construct to a named construct.

15. The method of claim 13, the transformation is lossless.

16. The method of claim 13, the mapping comprises one or more of a suppress field label, an introduce field label, a default mapping, a user customized mapping, and a mediating schema.

17. The method of claim 13, the mapping is based on one or more of a heuristic, an inference, a probability and machine learning.

18. A method that transforms constructs, comprising:
providing a construct to transform;
retrieving a mapping that facilitates construct transformation; and
utilizing the mapping to transform the construct.

19. The method of claim 18, the mapping comprises at least one of a suppress field label, an introduce field label, a default mapping, a user customized mapping, and a mediating schema.

20. The method of claim 18, the meditating schema transforms constructs to an intermediate representation at least one of before, during and after transforming the construct.

21. The method of claim 18, the received construct is a complex or simple construct and the transformed construct is a complex or simple construct.

22. The method of claim 18, the transformation comprises serializing a markup construct to an object construct.

23. The method of claim 18, the transformation comprises deserializing an object construct to a markup construct.

24. The method of claim 18, the transformation comprises persisting an object construct to a relational construct.

25. The method of claim 18, the transformation comprises restoring an object construct from a relational construct.

26. The method of claim 18, the transformation comprises publishing a markup construct in a relational construct.

27. The method of claim 18, the transformation comprises shredding a relational construct to markup construct.

28. The method of claim 18, the transformation comprises binding the received construct to a user interface, the received construct is one of an object construct, a markup construct, a relational construct and a disparate user interface construct.

29. A data packet transmitted between two or more computer components that facilitates transforming constructs, comprising:

obtaining a mapping filed comprising at least one of a set of suppress field labels, a set of introduce field labels and a mediating schema; utilizing the mapping file to transform a first construct to a second construct; and outputting the second construct.

30. A computer readable medium storing computer executable components to facilitate transforming constructs, comprising:

a component that receives a construct to transform;

a component that provides a mapping that facilitates construct transformation; and

a component that utilizes the mapping to transform the construct to a different domain space.

31. A construct transformation system, comprising:

means for determining a mapping between constructs; and

means for employing the mapping to transform a first construct to a second construct.